2006-01-1170

Analysis and Modeling of Servo Motor Control in Military Vehicle Control Systems

Anthony J. Torre
General Dynamics Land Systems

Christopher B. Mushenski
PM-MGV Common Systems, U. S. Army, TACOM-Warren

ABSTRACT

This paper covers the analysis of various concepts that are used to implement Servo Motor Control in military vehicular control systems. A survey of existing “hard real-time” servo controls and applicable design patterns is presented. These design patterns along with their critical parameters are identified and described. Potential solutions are created from combinations of design patterns and parameter choices. The solutions are modeled with plausible parameters to identify parameter sensitivities. As the modeling progresses, the solutions’ frequency responses, sensitivities and functional performance are also evaluated. This paper concludes with a summary of architectural guidelines.

INTRODUCTION

There is an increasing trend in recent years to use decentralized “hard real-time” control implementation in military and commercial vehicles. The driving forces for this include the following:

- Advances and cost reductions in network technology
- Demands for commonality
- Demands for higher reliability
- Improved limp home capabilities

The advances and cost reduction in network technology are real. The networks are now faster and more reliable than ever. Examples of this are given in the “Rotorcraft” presentation by Dale Johnson, reference [11], which shows the past and predicted evolution of network technology. The advances are continuous. Ethernet for example was 100 Megabits per second in 2000, reached 10 Gigabits per second in 2005 and is expect to reach 100 Gigabits per second in 2010. The possibilities of using remote input / outputs and remote actuators over the networks are now feasible.

The implementation however is not always straight forward and has a number of pitfalls especially with regard to latency and jitter. This paper addresses a few of the problems and some of the associated solutions.

Many solutions turn out to be concepts that have been under development for years in universities and research facilities. In other cases there are new areas to be researched and documented.

COMMON BUILDING BLOCKS

The demand for commonality comes from architectural desires to use common “building blocks”. It is thought that common units, when properly designed, can be used across the vehicle product line. While the use of “product line” development is used in commercial application, the commonality approach is strongly desired by the US Army. The reason for the emphasis on commonality is the desire to be able to take common parts from one vehicle and use them in a totally different control in another vehicle.

The most common division of systems is into the basic building blocks of input / output units, computing units, power control units and servo controllers. This vision of simplicity in the future is illustrated in Figure 1. Multiples of each building block type can be used to facilitate redundancy, “limp home” capabilities, expansion, and reconfiguration. The redundancy and the reconfiguration facilities provide the increased reliability needed for the survivability of the solders in the field.

Report Documentation Page			Form Approved OMB No. 0704-0188
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.			
1. REPORT DATE **10 FEB 2006**	2. REPORT TYPE **Journal Article**		3. DATES COVERED **10-02-2006 to 10-02-2006**
4. TITLE AND SUBTITLE **ANALYSIS AND MODELING OF SERVO MOTOR CONTROL IN MILITARY VEHICLE CONTROL SYSTEMS**			5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Christopher Mushenski; Anthony Torre**			5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **General Dynamics Land Systems,38500 Mound Rd.,Sterling Heights,MI,48310**			8. PERFORMING ORGANIZATION REPORT NUMBER **; #15522**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **U.S. Army TARDEC, 6501 E.11 Mile Rd, Warren, MI, 48397-5000**			10. SPONSOR/MONITOR’S ACRONYM(S) **TARDEC** 11. SPONSOR/MONITOR’S REPORT NUMBER(S) **#15522**
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**			
13. SUPPLEMENTARY NOTES **SAE WORLD CONGRESS 2006 COBO HALL, DETROIT, MI**			
14. ABSTRACT **This paper covers the analysis of various concepts that are used to implement Servo Motor Control in military vehicular control systems. A survey of existing ?hard real-time? servo controls and applicable design patterns is presented. These design patterns along with their critical parameters are identified and described. Potential solutions are created from combinations of design patterns and parameter choices. The solutions are modeled with plausible parameters to identify parameter sensitivities. As the modeling progresses, the solutions? frequency responses, sensitivities and functional performance are also evaluated. This paper concludes with a summary of architectural guidelines.**			
15. SUBJECT TERMS			

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

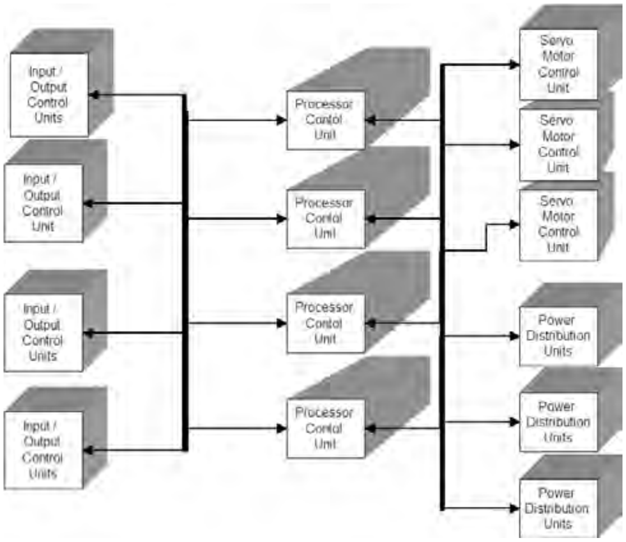

Figure 1 - Example Distributed Computer Control System

EXISTING TECHNIQUES

The following sections describe a survey of the real-time control techniques and explanations of each technique. The optimal solution for a specific problem will come from using a set of these techniques to accomplish the goals of the system.

SURVEY OF EXISTING INDUSTRIAL TECHNOLOGY

The goal of this survey was to establish what techniques were used in existing control system technology. The area of industrial controls and robotics were of special interest because of the similarities to military vehicle applications. The survey was generated primarily from researching Internet documents and university libraries. The documents that were the most helpful were Doctorial Theses (such as reference [4] and [9]) and technical studies from industrial control groups such as IAONA (Industrial Automation Open Networking Alliance) [10].

Considerable information was gathered from the European industrial control standards groups who were using some techniques that appeared to be more advanced than what was being considered in our initial efforts. This is especially true in the use of Ethernet Data Packet Packing to create high speed low latency data links.

Of interest was that the transmission of formatted data in the standard message data format was utilized in only two instances, EtherNet/IP and Modbus/TCP. In three cases Ethernet Frames were constructed by multiple controllers in a non-standard manner. The results of using the "multiple computer message packet construction" was fast data update rates. The fast data update rates came at the expense of non-standard hardware, which is required to support the multiple computer construction of message packets.

One of the other techniques that was widely used was the use "Time Division Multiple Access" (TDMA). This came in the form of either "isochronous message packets" or "time triggered protocol" (TTP). Table 1 shows the list of applicable techniques that were determined for common use, their benefits and drawbacks. To apply these techniques, a topology has to be implemented.

Global Clocks in Distributed Control Systems

In distributed computer control systems, there is a requirement to synchronize a master clock with remote clocks located on the distributed units. There are various standards used to do this synchronization. The Ethernet de facto default standard, from evaluation of the different protocols, was the global clock synchronization standard, IEC 61855. This standard defines a method to match remote clocks to the global clock with less than 1 microsecond of inaccuracy, which meets the requirements of time trigger protocols and time stamping of data. The accurate time stamping of data facilitates latency compensation of time critical data, which can then be used to improve the overall accuracy of reference data (e.g. position, speed or torque command data).

Goals of Servo Network Implementation

Our goal is to apply these techniques to develop servo system implementations that will meet the needs of platform protection, propulsion systems and survivability. To implement this properly requires the following characteristics:

- Determinism
- Minimized Latency
- Minimized Jitter
- No lost message packets
- "Highly Dependable" communications

Highly desirable attributes include the following:

- Guaranteed delivery of data packets
- Composability
- Minimum Weight
- Minimum Volume
- Reconfigurable
- High Reliability

Composabilty is the new word in this group. It is the ability to add additional control units to a data network without affecting the existing units. One test for this is to add a "babbling idiot" control unit, e.g. a continuously sending transmitter, to the network and verify that it has not affected other control units' operations. The truth of the matter is that composability is difficult to achieve with out the use of Time Division Multiple Access e.g. Time Triggered Protocol or Isochronous communications. These will be discussed later in the paper.

Table 1 - Applicable Real-Time Control Techniques for Control over Distributed Computers

Technique	*Pros*	*Cons*	*Common Existing Technology*
Highest Priority Processing	Functional for one high priority application	Does not work well on more that one high priority	Yes
Time Triggered Protocol	Composability, simplifies formal verification, and reduces latency when compared to Event Triggered protocol	Require complete knowledge of resource allocations. Requires special timer hardware and synchronization	Yes
Timed Task Activation	Minimizes latency	Requires task activation timers	Yes
Dedicated Dual Controller Hardware	Uses common LRUs	Has a minor size impact.	Yes
Single Controller Hardware	Minimum Latency and maximum control	Least amount of fail safe and reconfiguration	Yes
Latency Compensation	Significantly reduces effect of latency.	In practice will still leave a level of noise.	Yes
Data Packet Packing	Synchronizes by start of message packet.	Limited number of devices on the bus	Yes
Data Switches	Pseudo Deterministic, large number of units on bus.	Cost, weight and volume plus additional hardware.	Yes
Global Time Synchronization	Facilitates time stamping of data and events.	Requires a limited use of the network periodically to update the remotes to compensate for remote clock drifts.	Yes
Non-periodic control law	Eliminates jitter.	Not well supported by control theory.	No
Segmented topology	Limits the number of nodes on the bus to minimize interaction	Limits some of the opportunities for reconfiguration	Yes
Star topology	Has extensive re-configurability and back up	Causes issues with composability and weight	Yes

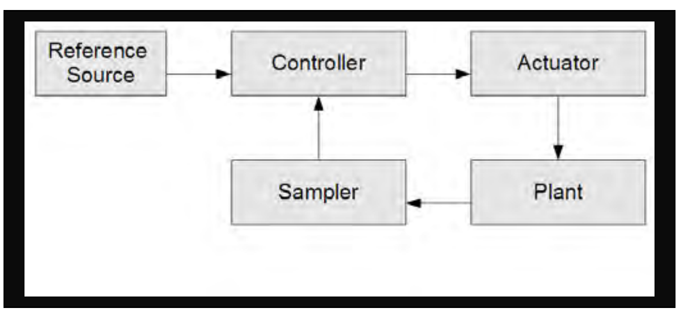

Figure 2 - Simple Control Loop

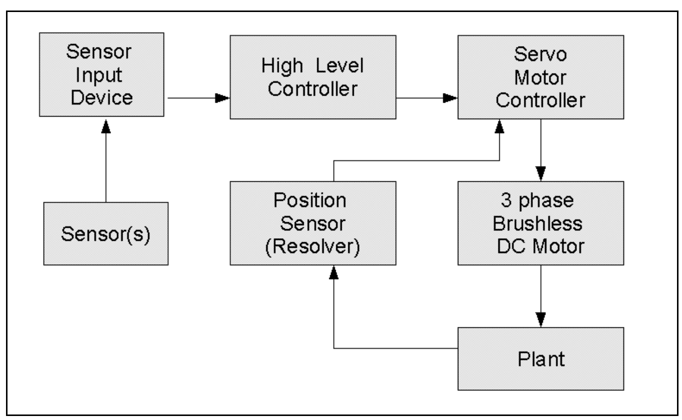

Figure 3 - Servo Control System Evaluation Model

Simple Models of the Systems

The basic control system is modeled in Figure 2 and shows a simple model of a closed loop control. In our study we utilize the distributed computer system, which requires a remote input-unit, high level computer processor and a servo actuator controller, a servo actuator, an actuator and a feedback sensor. This is shown in Figure 3. We will use a sensor input device, a high level controller, a servo motor controller and a 3 phase brushless DC motor with an attached resolver as our test case. The plant in this example shall be a simple inertial and friction load. The system will be modeled with ANSOFT Simplorer.

LATENCY CONTRIBUTORS

Because we are analyzing the control system, we want to identify the latency contributors. They are described in the following subsections.

The delays are modeled as they are observed. The processing delays will be modeled as pure propagation delays.

Sensor Input Processing

If we assume an event driven system then, the sensor input processing delays included are defined in Table 2.

Table 2 -Sensor Input Latencies

Identified Effect	Reasonable Value
Sensor Input processing	48 usecs
Scaling and packet building	16 usecs
Packet transmission Delay	64 usecs

The summary of these effects would be modeled as 128 microseconds (usecs) of pure delay. The packet sending time is normally a function of the loop time in event triggered systems. Since input processing is normally performed at the start of loop execution cycle and the transmission of serial message is an output function it is normally done at the end of the main loop. The maximum delay loop time is 500 microseconds and the minimum is the 128 microseconds. Our model shall use a message delay of 64 as the model in this event triggered implementation. The total worst case delay is 564 microseconds.

Note that one of the common mistakes by novice working on latency studies is to not include the normal software processing time. Since embedded software commonly uses the input / control / output architecture, it will most likely be used unless specified otherwise. To not include its timing is a serious mistake.

Having read the sequencing in the above paragraphs, one realizes various ways to improve it. If we time the tasks so that the data is read and processed just before the message is sent, it can substantially reduce the latency of the data. This is illustrated in Figure 4 and is referred to in the paper as “timed task activation.”

By adding a timer for sensor processor interrupts, we can reduce the latency between reading the data and sending the data. A reasonable estimate for the interrupt processing is dependent on the other activities in the sensor processor. A new latency value of 128 microseconds could represent the remote sensor delay and add it to the message packet propagation delay to get a total of 192 microseconds of delay.

Timed task activation does have a cost. It requires “time-triggered procedure-calling” facilities. Fortunately these are standard on real-time controllers under the names like “event processor array” and “timer processing units”.

High Level Controller

The purpose of the high level controller is to take the sensor inputs, process them and provide a reference to the servo motor controller. An example would be to take the handle information and inertia signals and speed of the traction drive motors.

The high level controller receives data from the joystick controller at rates of 500 microseconds. The high level controller can be run with 500 microsecond main loops also. There is a significant problem in that there is a lack of synchronization between the sensor input device and the high level controller. If one looks at the sensor input device and uses it as the time reference, the sensor input device sends the message packet, but since the main loops are not synchronized the packet will not be received until the input processing of the high level controller decides to read the input. This delay is equal to the main loop time of the high level controller. Figure 5 illustrates the worst case delay due to this lack of synchronization.

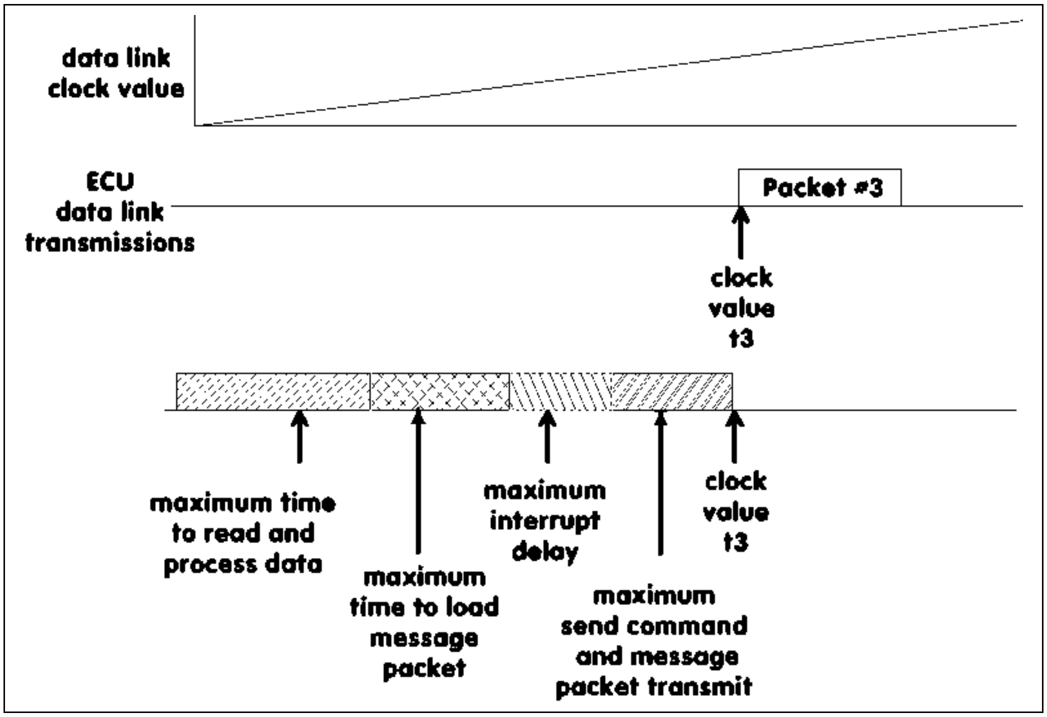

Figure 4 - Timed Task Activation

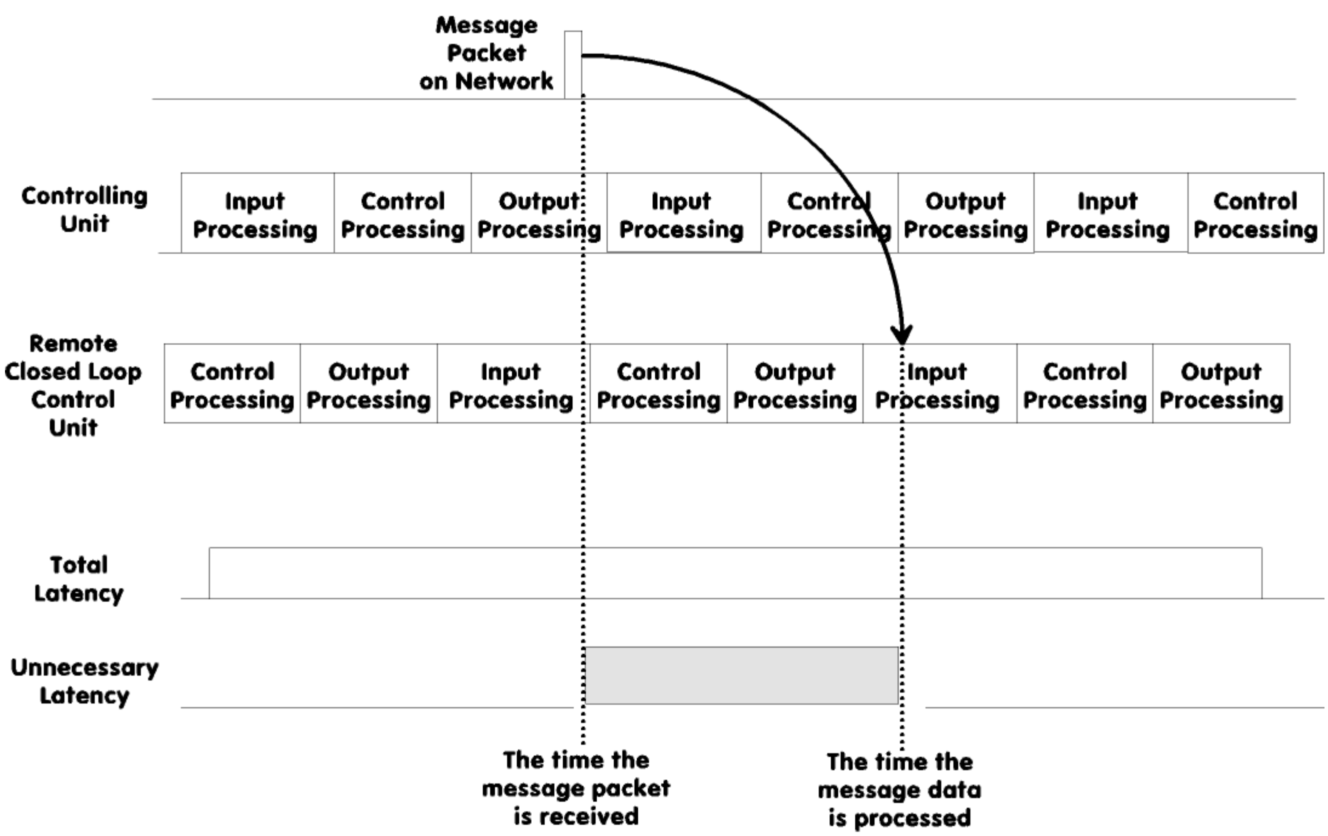

Figure 5 – Effect of Lack of Distributed Controller Synchronization

Table 3 - High Level Controller Delays

Identified Effect	Reasonable Value
Synchronization Delay	500 usecs
Packet storing	50 usecs
Packet processing	100 usecs
Reference Calculation	100 usecs
Scaling and packet building	16 usecs
Packet transmission Delay	64 usecs

Table 3 shows the high level controller delays. The processing delay is 500 microseconds for the high level control. The total delay is 1064 microseconds. Again the timed task activation (TTA) method can reduce this time.

There is also another technique that can be used to speed up processing. By processing all incoming data at the highest priority possible, and to process the input data as soon as possible, and the associated calculations as soon as possible, the data transfer can be expedited. The problem with this technique is it only works on one device at a time and does not work well in concurrent processing situations.

The process utilization on the high level controller is not extensive. The updates required are in the range of 500 microseconds (for wheeled vehicle steering) to 100 milliseconds (for thermal management systems). Attempts to parallel control tasks in the high level controller quickly lead to resource bottle necks unless certain scheduling techniques, such as Time Trigger Protocol, are used.

<u>Servo Motor Controller</u>

The servo motor controller receives a reference from the high level controller. The reference is used to close the loop based on position, speed or current. The servo motor controller performs these tasks by following specific timing ranges for fast processing as shown in Table 4:"

Table 4 - Typical Timing Ranges per Control Type

Reference	Control Loop Speeds
Position	1 to 10 milliseconds
Speed	100 to 1000 microseconds
Torque	10 microseconds to 100 microseconds

In order to process incoming reference quickly the servo motor controller can process the incoming data in the same interrupt loop that stores the packet. While this violates normal software processing rules, it significantly reduces delay times and increases stability. This method is referred to as “highest priority processing”. Table 5 shows the servo motor controller delays.

Table 5 -Servo Motor Controller Delays

Identified Effect	Reasonable Value
Packet storing	50 usecs
Packet processing	100 usecs
Reference Calculation	100 usecs
Output	10 usecs

<u>Single Controller Processing</u>

If one studies and analyzes the throughput requirement, it is clear that the High Level Controller and the Servo Motor Controller can be integrated into one unit. The throughput of the Servo Motor Controller is much higher than the High Level Controller therefore the processing can easily be absorbed by the Servo Motor Controller. The unification to the two controllers greatly reduces the latency since the main loop delays are minimized. One can also pull the sensor input processing into the high level control and reduce the latencies and optimize the sequencing.

ADDITIONAL OPTIMIZATION TECHNIQUES

This section describes techniques to be applied to the whole system. They will be applied and evaluated later in the paper. Some have already been mentioned but are elaborated here.

<u>Synchronization of Commands</u>

Serial Communications can be implemented in four approaches. They are as listed:

- Event triggered
- Time triggered
- Isochronous
- High Priority Processing

Event triggered has been used as the “norm” for many years and is still utilized on most military and commercial vehicle applications. For the most recent safety, high dependability and high reliability systems, the isochronous approach is utilized. The isochronous technique, a form of TDMA, has substantial advantages above the event triggered technique which include guaranteed data slots and composability.

The emerging approach is time triggered protocol. The time triggered protocol has been under development for over 20 years by the Technical University of Vienna under sponsorship from the European Union. The time triggered technique also has the advantages of guaranteed data slots and composability.

The emerging technique of choice appears to be "concurrent time triggered protocol and event trigger protocol". The rational appears that time trigger protocol is desired to schedule processing, but the event triggered protocol is required to report the unexpected. Flexray is one of the emerging protocols that utilizes both the time triggered protocol and the event triggered protocol.

High priority processing is utilized but effective only for a limited number of instances. This is perhaps one of the reasons that so many servo controllers are dedicated control units. This is in addition to the fact that the number of resource conflicts that occurs, increases with each additional implementation.

The following subsections shall elaborate further each technique. The techniques will be applied to various systems for evaluation later in the paper.

Event Triggered Protocol

The "event triggered" technique requires the "control unit" to send out data at a time designated by an event. The event is normally arbitrary and not synchronized with other events. The problem is that if more than one controlling unit or other transmitting units attempt to send a message at the same time on the data link, then a conflict for access to the data link occurs. Solutions to this problem include the use of message priorities (CAN) and using data switches (Ethernet). The root of the problem, as state in [1] is that "temporal control in an event triggered system depends on the behavior of all controllers within the system". The event driven systems will not have the ability to add new units to the data bus without the possibility of affecting the whole network.

Time Triggered Protocol

Time triggered protocol initiate tasks by triggering on globally synchronized timers. This method operates by giving each controller an amount of time to in which they give exclusive sending rights to each electronic control unit on the data bus. This then provides a "guaranteed data slot" for the specific control unit that is transmitting.

To synchronize the control units global timers, delays in messages transmission are established and compensated for. The main global clock then is used to synchronize the local global clocks in the control. As previously mentioned, the most commonly used technique for this synchronization is specified by IEC 61588. One of the advantages of the time triggered protocol is that it can make a standard Ethernet-based system truly deterministic without having to add the weight, space and cost of data switches to the system.

If one combines the time triggered protocol with a timed task activation and then sequences the supporting tasks so that the minimum possible latency is achieved, the only additional delays that are required are the time for concurrent activity interrupts. The problem with the concurrent activity interrupts is that the vary independent of the control loop tasks. One way to control it is to issue strict requirements on the number of interrupts and the amount of execution time consumed over a fixed period in which the control system will operate.

Be aware that time triggered architectures are less flexible than event driven systems but are less difficult to analyze and test. Specifically designated allocation slots must be assigned for the time triggered architecture where as the event trigger architectures devices can just be plugged in and be working. However, the ability to analyze and test more can easily be worth the extra effort. This is especially true in "highly dependable systems" such as platform protection systems.

Isochronous Systems

Isochronous Systems are "time division mutual access" protocols like time triggered protocols. The main difference is in the timing method. In the Isochronous System, a starting packet is periodically sent to trigger when messages are to be sent. As an example IEEE 1394b uses an 8 kHz update rate to trigger message packet to be sent in a predetermined order based on control unit placement in the tree topology. This technique simplifies the timing synchronization.

Figure 6 is an example time link, which illustrates that the isochronous packets and the event driven packets can be concurrently utilized. This provides for control packets and for event triggered packets such as fault messages.

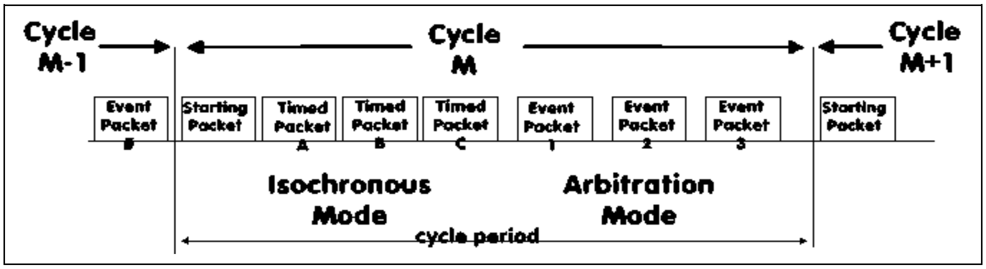

Figure 6 - Isochronous Mode

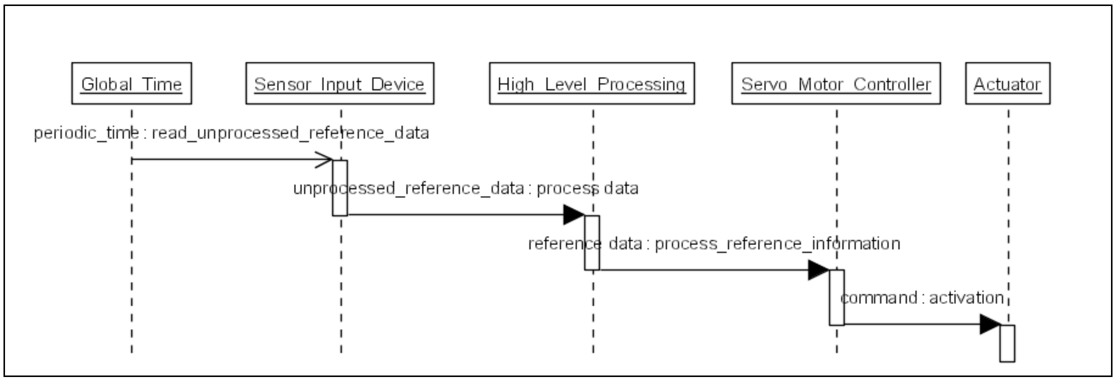

Figure 7 - High Priority Processing

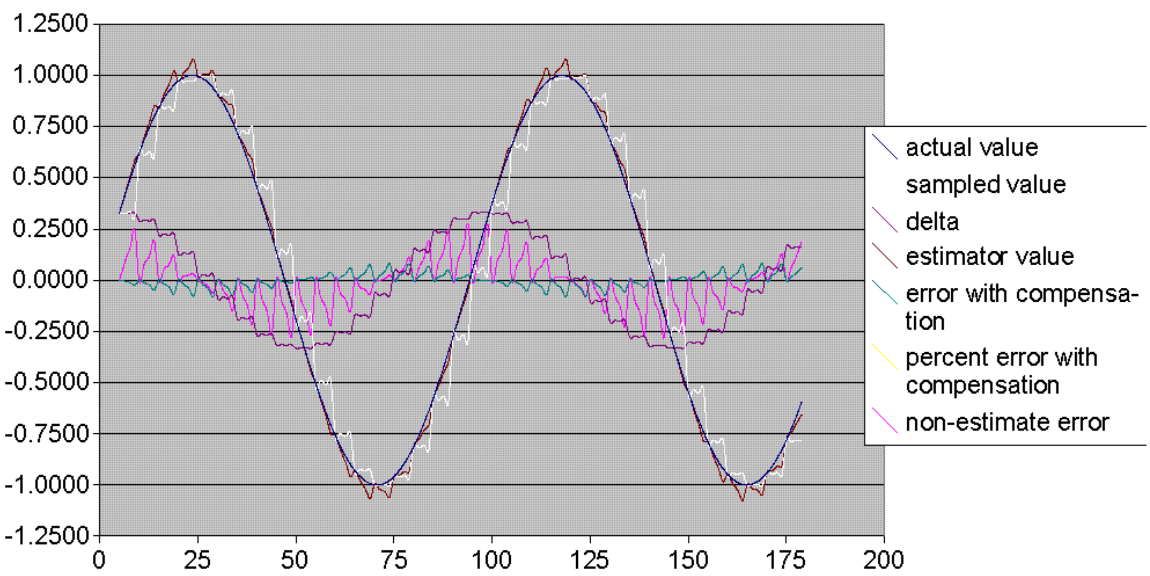

Figure 8 - Time Compensation Simulation

High Priority Processing

High priority processing can be used to control the update sequencing. This sequence starts with the remote input unit reading the signal at a periodic rate and sending it to the high level controller. The high level controller then immediately calculates the reference signal (position, speed or current) and immediately sends the reference to Servo Motor Controller. The Servo Motor Controller processes the input of the reference parameter immediately after reception. This is illustrated in Figure 7. Using the high priority approach, the timing approaches the single unit controller. There is however considerable jitter that is developed by variations caused by process flow variation and higher priority interrupts. The correction for this problem is to assign the processing event specific times to occur. This again leads back to a "time division multiple access" techniques e.g. "time trigger protocol" or "isochronous" packets. Further improving a system with TDMA then requires timed task activation.

Latency Compensation

Latency compensation can be accomplished by anticipating the value at the time it is needed. The most simple technique for doing this is adding the last read value to the "parameter rate of change times the difference between the measured time and the need time. This is shown in pseudo code as the following:

```
rate_of_change = parameter_value_k
                        – parameter_value_k-1

time_difference = time_of_measured_parameter_value
                        – time_of_parameter_value_use

final_parameter_value = parameter_value_k
                    + (rate_of_change * time_difference)
```

Simulation results using a sampling rate of 24 times the highest frequency shows a reduction in the standard deviation of noise by 3.3 times. This is shown graphically in Figure 8. The white line shows the uncompensated data. The ramping dark line that is both the highest and lowest sign show the compensated data. The original signal is the signal wave in dark ink.

Latency compensation implementation is possible using various estimators [12]. The technique shown here is the crudest but the easiest to implement. The technique of latency compensation is widely used and can be applied to sensor processing such as resolvers as well as compensation for actuators. Actuator latency compensation can be implemented by using the time the control actuator fires as the current time, instead of the time that the control law is calculated. This is helpful in controls that use solenoids and have dead bands.

One target for applying this should be with internal combustion air/fuel controls where there is a temperature dependent time of combustion that is on the order of 20 milliseconds. The results should be a smoother and more stable idle.

Jitter Compensation

Jitter compensation can be implemented by adding the actual timing deltas into the control law. That is to replace the constant period time with a calculated value for the delta time. The use of a standard period value is utilized and simulated by existing theory and tools. The use of period control is not the only technique but is the most developed on. By replacing the period "h" in our control laws with the delta time, yields a far more jitter tolerant control.

A simple example of the current technology on an estimator of a derivative of "x" is:

$$dx/dt = (x(k) – x(k-1)) / h$$

Where h is the constant period.

A more jitter tolerant version is the equation below:

$$dx/dt = (x(k) – x(k-1)) / (t(k) – t(k-1)$$

To implement the jitter tolerant control and still minimize the risks, one can "bound" the calculation by a percentage of the period e.g. 10% of period. This limits the impact on monitoring tools and simulations. In fact the recommended approach here would be to develop the control laws with the fixed periodic approach and then replace the fix time controls with ones that use the variable time control laws. The result would be a more jitter tolerant control. The jitter tolerant approach would allow hard real-time controls to function in an environment where execution timing resource limitations are starting to appear. This resource limitation can occur when parallel programs are running on the same processor and the number of external interrupts are bounded but not precisely known.

Be aware that there are many jitter contributors in Single Program on a Single Processors. These include the following:

- High priority interrupts
- Missed Caches
- Interrupt lockouts

When parallel processing is used, these jitter contributors are greatly increased. The other programs running concurrently need to be constrained by design guidance so that high priority interrupt times are managed properly. The concurrent interrupts need to be limited in time and number.

SOLUTION TEST CASE SETS

The goal of this paper is to recommend solutions that are composed of the previously identified techniques that are used and that are applicable. To do this, we established the following Solution Test Case Sets as shown in Table 6 and evaluated them on an ANSOFT Simplorer Simulation depicted in Figure 9.

Table 6 - Solution Test Case Sets

Test Case Label	Subsystem Configuration	Data Switch Used	Improved by Jitter Compensation	Improved by Latency Compensation	Time Division Multiple Access
All-in-One	Single High Level and Servo Motor Controller	Not required	Very slightly	Very slight	None
Event Distributed GbE	Distributed Control with Event Trigger	Required	Significantly	Significantly	None
Distributed With TTP and TTA	Distributed Control	Not required	Very slightly	Slightly	TTP and TTA
Distributed with Isochronous Communications	Distributed Control with Time	Not Required	Slightly	Slightly	Isochronous.

The solution "all-in-one" is an element in a federated system, which performs all the high level servo control (e.g. reference development) and all the control is in one single control unit. This unit communicates with other units for supervisory commands and status reporting. Other than that the 'all-in-one' solution is a time optimized and totally cohesive control unit.

The advantage of the "all-in-one" controller is that it can optimize each step of its operation with little or no input from other controllers. There is the assumption in this approach that if more than one processor inside the "all – in-one" controller is used, that communications between the processors have no additional significant delay.

The "Event Distributed GbE" configuration uses "event driven" serial communications over Gigabit Ethernet. This is commonly used for soft real-time applications and requires the use of a "data switch". This approach is highly regarded by Professor Decotignie in [7].

The "Distributed with TTP and TTA" is a distributed system of control units with global time synchronized remote clocks. The associated control units communicate over a Time Triggered Protocol. The units also minimize the latency by scheduling input processing tasks at the appropriate time before they are needed.

The "Distributed with TTP and TTA" approach also allows the event driven operation for diagnostics and other statuses. The "Distributed with TTP and TTA" approach appears to be the solution of choice for evolving trend for automotive applications. This configuration is embedded in Flexray communications. The Flexray standard has been evolving in for several years by multiple automotive companies and its specification has now been released.

The "Distributed with Isochronous Communications" solution is distributed units with isochronous communications such as IEEE 1394. This configuration has been the solution of choice for highly dependable systems and safety related systems. It has been used in industrial control and to a limited extent in automotive safety related systems.

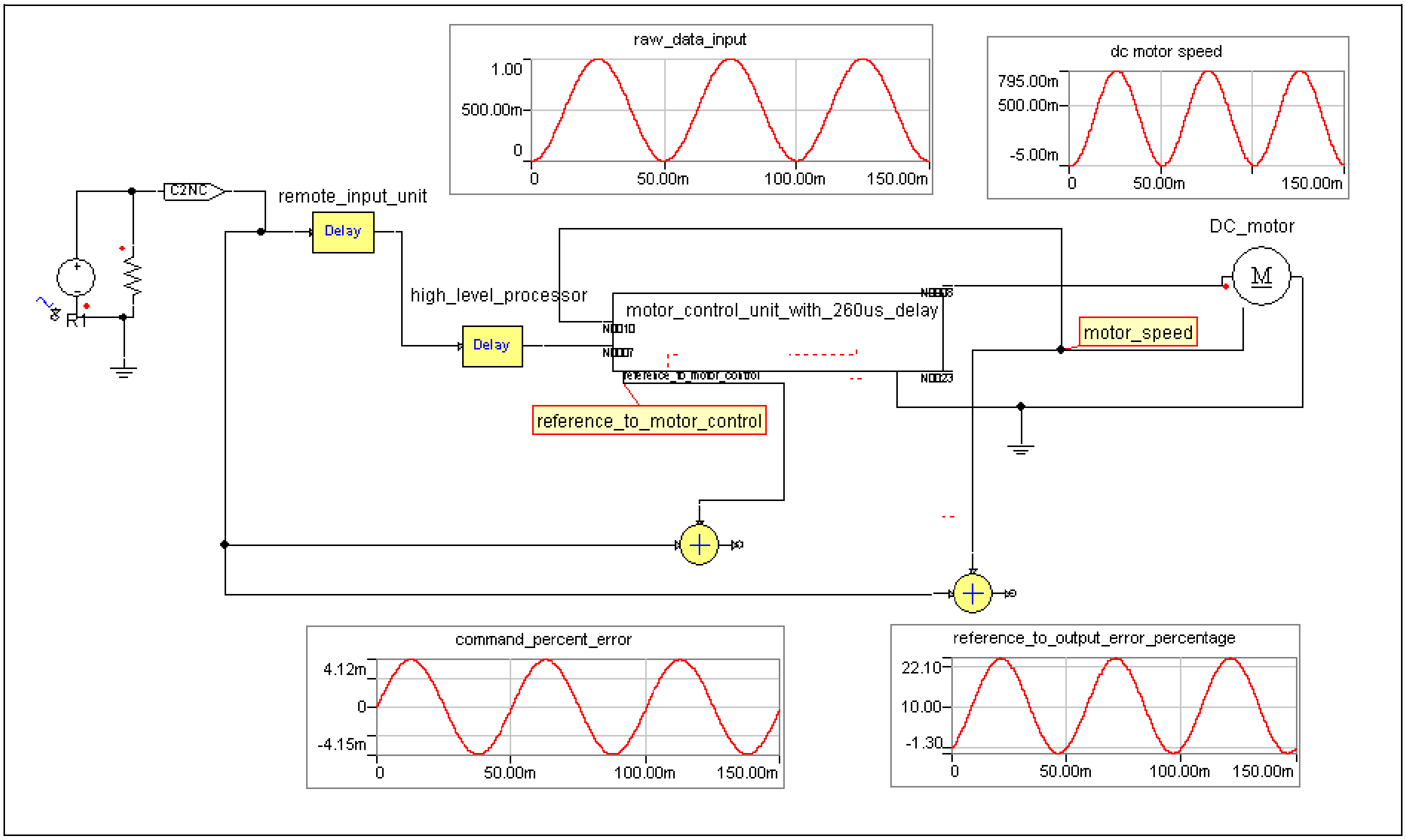

Figure 9 - Simplorer Model of Servo Control System

Table 7 – Results of Simulations

Test Case Label	Sensor Processing Delay (usecs)	High Level Control Delay (usecs)	Servo Motor Controller (usecs)	Reference Latency Total (usecs)	Percentage Reference Signal Error on 20 Hertz Signal
All-in-One	0	0	160	160	1.0%
Event Distributed GbE	564	1000	260	1824	11.5%
Distributed With TTP and TTA	128	266	260	654	4.0%
Distributed with Isochronous Communications	192	391	260	843	5.3%

In Table 7, the four test cases were evaluated with and without Jitter and Latency Compensation. This gives eight test case sets in which to the select the best configuration for the system. Simulation was provided for all the cases. A formal trade study is then performed using the data gathered.

The results of the simulations are shown in Table 7. The focus was not made on latency error of the reference signal which was the section that varied the most due to our solution configuration changes.

The first observation is that the "all in one" dedicated controller is still the best solution based on only error. This is no surprise since this is the legacy system and the configuration where timing and sequencing can be ideally coordinated.

The second observation is that the distributed computer system using time triggered protocol and timed task activation is the second best solution with 4% error. Additional time latency compensation to this solution reduces the error to 1.4% which is closer to the "all in one solution".

TRADE STUDY REQUIREMENTS

To come up with an appropriate solution for one's application requires a formal trade study with metrics similar to those shown in the Appendix. This is beyond the scope of the paper and will not be discussed further.

GUIDELINES

The following guidelines have been identified for use by organizations attempting to use distributed computer systems for servo control systems.

- Use latency compensation to significantly reduce the effect of latency delays.
- Use networks which allow use of both time triggered and event triggered communications.
- Use time variable control laws to remove the effects of jitter.
- Use a single controller for time critical control systems where applicable.
- Where redundancy is required on time critical control systems, use a second single control unit.
- Use time triggered protocol and timed task activation on control systems implemented on distributed computers.
- Unless using time triggered protocol and timed task activation, avoid concurrent processing of control system software on a single computer.
- Use time triggered protocol to remove the cost, space and weight of data switches on networks.
- When using concurrent processing, bound the number and the execution time of interrupts to reasonable levels.

CONCLUSION

The conclusions are as follows:

- A distributed computer control system design has latency and jitter contributors that are not always obvious. New designs should have proper analysis done before developing the top level design.
- Latency compensation should be considered for use in distributed computer control systems.
- Time Triggered Protocol can reduce the weight and volume from subsystem by removing switches and hubs.
- Event driven data bus communications has inherent delays that are significant and variable.
- Time variable control laws should be considered to remove jitter.
- The best solution from the timing performance is the federated "all-in-one" configuration.
- Additional control techniques (such as TTP, TTA, latency compensation and jitter compensation) need to be used on fast control systems which are implemented over distributed control units.

While much of the distributed computer control system technology is familiar; there are new associated areas then need to be explored. To be complete, additional effort needs to be made in this area. Different techniques are required to implement the high speed control systems over distributed control units.

REFERENCES

1. Thomas Führer, Bernd Müller, Werner Dieterle, Florian Hartwich, Robert Hugel, Michael Walthe; Robert Bosch GmbH. "Time Triggered Communication on CAN"
2. TTA Group, "TTP – Frequently Asked Questions", 2004
3. Jurriaan H. Wijnberg. "Prototyping the real-time 'HOTnet' network protocol on Ethernet"; Master's Thesis, University of Twente, April 2002
4. Ola Redell. "Response Time Analysis for Implementation of Distributed control Systems", Doctorial Thesis, Royal Institute of Technology, Stockholm Sweden, 2003
5. Hubert Kraut. „Signalverarbeitung mittels eines Neuronalen Netzwerkes für einen Smart Sensor", Thesis, Vienna University of Technology, February 2003
6. "CAN in Automation" organization. "CANopen Product Guide 2003".
7. J.-D. Decotignie. "Real-Time System II Real-Time Networking Ethernet", CSEM Centre Suisse d'Electronique et de Microtechnique SA, 2005
8. Dr. Dirk Janssen. "Als Antriebsbus geeignet, weil...", Beckoff, March 2005
9. Johan Nilsson. "Real-Time Control Systems with Delays", Doctorial Thesis, Lund Institute of Technology, Sweden, 1998.
10. Industrial Automation Open Networking Alliance. "IAONA Handbook Industrial Ethernet in third addition".
11. Dale Johnson. "US Army on Rotorcraft Technical Architecture", AATD.
12. Bo Lincoln. Jitter Compensation in Digital Control System, Department of Automation Control, LTH, Lund Sweden, March 2003

CONTACTS

Anthony J. Torre
General Dynamics Land Systems
38500 Mound Road
Sterling Heights, MI 48310
Email: torrea@gdls.com

Christopher B. Mushenski
PM FCS (BCT); PM MSI; PM-MGV Common Sys
SFAE-GCS-FCS-E, M/S 515
Warren, MI 48397-5000
Email: mushenc@tacom.army.mil

DEFINITIONS

Controlling unit: The computer based unit that performs the main algorithmic processing.

Composability: The guarantee the modifications or the addition of new functions will affect only specified systems.

Deterministic: An attribute of systems whose behavior is specified without probabilities (other than zero or one) and predictable without uncertainty once the relevant conditions are known. Deterministic systems leave nothing to chance.

Global Timed: A globally synchronized time base which is available for all nodes of the network with a precision which fulfills the real-time requirements of the application.

LRUs: "Line Replaceable Units" These are units that are replaced in the field as complete assemblies.

Remote Device: This is the remote I/O device that is used for reading and writing to remote devices.

TDMA: Time Division Multiple Access

TTA: Timed Task Activation

TTP: Time Triggered Protocol

Time Triggered Protocol: Time triggered communications on the data communications network.

Timed Task Activation: Time triggered operations at the application Level

APPENDIX

Example Metrics for the full evaluation of network configurations

Grouping	***Metric***	***Weight***	***Metric Rating Designation***
Cost	Operating and Sustainment Cost / Life-Cycle Cost	10	Constructive
Highly Dependable	Composability	8	Constructive
Highly Dependable	Determinism	10	Constructive
Highly Dependable	Guaranteed delivery of data packets	5	Constructive
Highly Dependable	High Reliability	8	MTBF
Highly Dependable	No Lost Messages	7	Constructive
Highly Dependable	Reconfiguration	7	% of units reconfigurable after failure
Performance	Error with fixed frequency signals	10	Corresponding error
Performance	Minimal Jitter	10	Corresponding error
Physical Implementation	Volume	10	Liters
Physical Implementation	Weight	10	Kilograms
Reuse	Common Units	7	% of common units
Reuse	Replaceable controller in same product line	10	% of units applicable for scavenging
Risk	Schedule Risk / Technical Risk / Cost Risk	8	Constructive
Testability	Ability to Diagnose	6	Constructive
Testability	Mean Time to repair	6	Constructive